REMARKS

Claims 125-179 stand in the application. These claims correspond respectively to clams 197-251 added by the Amendment filed July 12, 2002. It is noted that in the office action mailed September 26, 2002, the examiner has indicated that claims 70-75, 90-101 and 110-113 are pending, but it the attention of the Examiner is directed to the first page of the Amendment filed July 12, 2002 where applicants requested cancellation of claims 70-75, 90-101 and 110-113. Thus, the only pending claims are claims 125-179. It is assumed that no excess claim fees have been charged corresponding to the inclusion of claims 70-75, 90-101 and 110-113.

Submitted herewith as requested by the examiner is a copy of page 80 which was included with the application as originally filed.

Claims 125-144 and 165-179 have been amended to address the rejection of those claims under the second paragraph of 35 U.S.C. 112. The amendments are believed self-explanatory, and withdrawal of the rejection is respectfully requested.

At the bottom of page 3 of the office action, the examiner has indicated that claims 90-101 and 145-164 appear to be allowable over he prior art of record. Claims 90-101 are not pending in the application, but have been replaced with claims 145-156, so that the statement at the bottom of page 3 is essentially a statement that claims 145-164 are patentable. While this indication of patentable subject matter is appreciated, it is noted that all claims pending in the application correspond to or are dependent from claims indicated in a previous office action as being patentable over the prior art, and no prior art has been or is now applied against any pending claims. Accordingly, allowance of all claims is believed in order.

Requirement for Information

Applicants have already brought to the attention of the examiner, either in the specification itself or in the Information Disclosure Statement filed October 5, 1999, prior art of which applicants were aware which might be material to the examination of the application. It is assumed that the Requirement for Information such art need not be discussed again hee.

Publications Relied Upon to Draft the Claimed Subject Matter.

The present application filed March 23, 1999 was prepared in reliance on (1) provisional applications nos. 60/079,100 filed March 23, 1998, and 60/096,909 filed August 17, 1998, (2) internal notes of and discussions with the inventors, and (3) the publications already cited at lines 18-26 of page 20 of the present application. The first-mentioned provisional application was prepared from press releases as is self-evident from the provisional application. No further publications were relied on to prepare the second-mentioned provisional application, and no further publications were relied upon to draft the present application as filed on March 23, 1999.

Names of Products or Services that Have Incorporated the Claimed Subject Matter

Products known to have incorporated the claimed subject matter include the "Web Automation Server" available as of December 1997 or January 1998, and subsequent products known as the "B2B Integration Server" and "Integration Server." The "Web Automation Toolkit" version 2.1 included XML-RPC sometime between December 1997 and March 1998.

Publications authored or co-authored by applicants which describe the disclosed subject matter of XML as used with Remote Procedure Calls

Chapters 8 and 38 of The XML Handbook, cited at page 20 of the present application. The best information the undersigned has been able to obtain indicates that this was published on or about June 26, 1998. The examiner is apparently already in possession of this publication, since Chapter 8 was cited by the examiner in the office action mailed September 26, 2002.

"WIDL", a speech and accompanying Power Point presentation delivered by Joseph Lapp at XML '98 on November 17, 1998.

Mr. Lapp also delivered a speech the B2B Integration Server, believed to be in late 1998 or early 1999. The undersigned will supplement this response if further relevant details regarding that speech can be located.

The remaining inventors may have given talks during 1998 and 1999 directed to or including XML-RPC, but it is not believed that any talks constituted or were accompanied by publications.

Applicants have located a posting by Joseph Lapp to XML.org dated September 22, 1998, and the undersigned is including a copy here without any concession as to whether or not this would constitute a publication. Applicants have not kept records of such postings, and have not conducted a search of the internet to see if any more can be found.

The undersigned has included a copy of a document entitled "What is XML" downloaded from applicants' web site on February 20, 1999, but perhaps posted to the web site as early as October of 1998.

The undersigned has not searched for or included in this response any publications which may have occurred subsequent to March 23, 1999.

Specific Improvements of the Claimed Subject Matter of Independent Claims

The independent claims standing in this application are claims 125-130, 145-156 and 165-172. An improvement reflected in all of these claims is the use of a markup language-based message encoding which realizes a number of advantages, e.g., the advantage described at the top of page 11 of the specification. Specific claim language reflecting this improvement is the use of a markup language-based message encoding wherein message elements have element type indicators" selected from a set. The claims then vary in reciting different characteristics of the message elements which can be designated by these indicators. (The claims have been editorially amended for consistency to use the term "element type indicators", of which the "element type names" used in XML would be an example.)

Claim 125 focuses on the operation of second machine in generating and transmitting the RPC, and claim 126 focuses on the operation of the first machine in receiving and acting on the RPC. In addition to the basic concept of using a markup language-based message encoding as described above, these claims reflect the advantage of the invention whereby the message itself identifies the container data structures as container data structures (i.e., elements which may contain other elements), and recipients of the message need no prior knowledge that the container structures will appear in a given message. Specific claim language reflecting this improvement is the use of at least two element type indicators which designate elements containing other elements from the same encoding group.

Claim 127 focuses on the operation of second machine in generating and transmitting the RPC, claim 128 focuses on the operation of the first machine in receiving and acting on the RPC, claim 129 focuses on the operation of the first machine in receiving, invoking and replying to the RPC, and claim 130 focuses on the operation of the second machine in receiving the reply. In addition to the basic concept of using a markup language-based message encoding as described above, these claims reflect the ability to have message constituents (e.g., VALUE elements) that convey the information content of the message separately message constituents (e.g., RECORD) that convey information about how that content is organized in the message. Specific claim language reflecting this improvement is the use of a set of element type indicators which includes a first type indicator designating an element containing data, and a second type indicator designating an element containing a set of further elements.

Claim 145 focuses on the operation of second machine in generating and transmitting the RPC, claim 146 focuses on the operation of the first machine in receiving and acting on the RPC, claim 147 focuses on the operation of the first machine in receiving, invoking and replying to the RPC, and claim 148 focuses on the operation of the second machine in receiving the reply. In addition to the basic concept of using a markup language-based message encoding as described above, these claims reflect the ability to designate in the message itself a data element as being a multi-dimensional array where the message indicates the number of dimensions and where each nesting level corresponds to a dimension of the array. This is illustrated in the examples at pages 49-50 of the specification where the 80/20 RPC expression uses nesting levels to represent array dimensions. One of the advantages of this aspect of the invention is described in the paragraph bridging pages 50-51 of the specification. Specific claim language reflecting this improvement is the use of an array type indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, and the message including at least one data item which is a multi-level nested array element where each nesting level corresponds to a respective dimension of said array element.

Claims 149-156 are directed to the designation and representation of dimensional arrays as described at pages 49-51 of the specification. In addition to the use of a markup language-based message encoding as discussed above, these claims reflect the feature of the invention

whereby for array type elements the size and the element type within the array can be specified using optional DIMENSION and TYPE attributes, e.g., as described at lines 3-12 of page 45 of the specification.

With respect to claims 149-152, note that in the examples given at pages 49-50, the label indicates the number of dimensions (i.e., the value of n) but does not indicate the size of each dimension. Instead, the size of each dimension is the number of elements within an ARRAY element. For example, the first nesting level which corresponds to an array dimension includes two ARRAY elements and is therefore of size = 2. Each of those first-level ARRAY elements includes two more ARRAY elements at the next nesting level, so the size of the second dimension for each ARRAY element is 2. Each second-level ARRAY element includes two data elements so their size is 2. Specific claim language reflecting this improvement is the language reciting that the label associated with the array type element designates it as an n-dimensional array but not indicating a size for each of said n dimensions.

With respect to claims 153-156, note the example at page 50 where for each ARRAY element at the first nesting level, the elements contained in that ARRAY element are both ARRAY elements, and for each ARRAY element at the second nesting level, the elements contained in that ARRAY element are both VALUE elements. As described at lines 3-12 of page 45 of the specification, the element type of the immediate children of the ARRAY element can be designated using an optional TYPE attribute. Since the TYPE designation is for an entire array, this will require that all immediate children elements of the array have the same type. This leads to more efficient processing of the content of the array element, as noted at line 8-10 of page 45. Specific claim language reflecting this improvement is the last two lines of each claim reciting that the encoding requires that all data items contained within said array as direct children have the same type as one another.

Claim 165 focuses on the operation of second machine in generating and transmitting the RPC, claim 166 focuses on the operation of the first machine in receiving and acting on the RPC, claim 167 focuses on the operation of the first machine in receiving, invoking and replying to the RPC, and claim 168 focuses on the operation of the second machine in receiving the reply. In addition to the basic concept of using a markup language-based message encoding as described

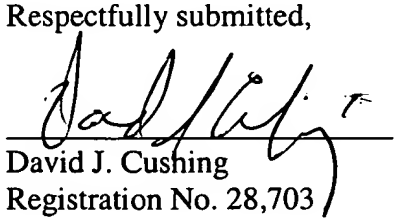
above, these claims reflect the ability to associate an element with an ID value and then have another element refer to that element simply by specifying its ID value. Specific claim language reflecting this improvement is the use of first and second element type indicators (e.g., RECORD and OBJECT), wherein said message associates an element having said first type with an ID value, and wherein the message includes an element having the second type indicator (e.g., OBJECT) which specifies the ID value.

Claim 169 focuses on the operation of second machine in generating and transmitting the RPC, claim 170 focuses on the operation of the first machine in receiving and acting on the RPC, claim 171 focuses on the operation of the first machine in receiving, invoking and replying to the RPC, and claim 172 focuses on the operation of the second machine in receiving the reply. In addition to the basic concept of using a markup language-based message encoding as described above, these claims reflect the ability to include in the message itself indications that certain message elements are placeholder elements. The advantages of being able to use placeholder elements are discussed at page 44 in the context of the OBJECT and NULL element types. Specific claim language reflecting this improvement is the inclusion in the group of type indicators of at least one placeholder element type indicator (e.g., NULL or OBJECT) that designates a placeholder element which represents the absence of data.

With the amendments made above, it is believed that all claims are in condition for allowance, and a Notice of Allowance is respectfully requested.

If there are any issues remaining which the examiner believes could be resolved through an Examiner's Amendment or a Supplemental Response, the examiner is kindly requested to contact the undersigned attorney at the local exchange indicated below.

Respectfully submitted,



David J. Cushing
Registration No. 28,703

SUGHRUE MION, PLLC
2100 Pennsylvania Avenue, N.W.
Washington, D.C. 20037-3213
Telephone: (202) 293-7060
Facsimile: (202) 293-7860
Date: January 15, 2003

APPENDIX
VERSION WITH MARKINGS TO SHOW CHANGES MADE

125. A method of invoking a service at a first machine from a second machine, comprising the steps of generating a service invocation request at said second machine using a markup language-based message encoding, and transmitting said service invocation request from said second machine, wherein said message includes plural elements and wherein all of said elements have element type ~~names~~indicators selected from an encoding group having a predetermined number of members, with at least two of said members ~~(e.g., RECORD and ARRAY)~~ designating elements containing other elements having element type ~~names~~indicators belonging to said group.

126. A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request generated at a second machine in compliance with a markup language-based message encoding, wherein said message includes plural elements and wherein all elements in said message have element type ~~names~~indicators selected from an encoding group having a predetermined number of members, with at least two of said members ~~(e.g., RECORD and ARRAY)~~ designating elements containing other elements having element type ~~names~~indicators belonging to said group; and

invoking said service in response to said request.

127. A method of invoking a service at a first machine from a second machine, comprising the steps of:

generating a service invocation request message at said second machine in compliance with a markup language-based message encoding, wherein said message includes plural elements and wherein all elements in said message have element type ~~names~~indicators selected from an encoding group having a predetermined number of ~~element type names~~members, including at least a first ~~(e.g., VALUE)~~ element type

~~name~~indicator for designating an element containing data, and a second ~~(e.g., RECORD)~~ element type ~~name~~indicator for designating an element containing a set of ~~children~~ elements having element type ~~names~~indicators selected from said group; and

transmitting said message .

128. A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding, wherein said message includes plural elements and wherein all elements in said message have element type ~~names~~indicators selected from an encoding group having a predetermined number of ~~element type names~~members, including at least a first ~~(e.g., VALUE)~~ element type ~~name~~indicator for designating an element containing data, and a second ~~(e.g., RECORD)~~ element type ~~name~~indicator for designating an element containing a set of ~~children~~ elements having element type ~~names~~indicators selected from said group; and

invoking said service in response to said message.

129. A method of invoking a service at a first machine, said method comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding, wherein said message includes plural elements and wherein all elements in said message have element type ~~names~~indicators selected from an encoding group having a predetermined number of ~~element type names~~members, including at least a first ~~(e.g., VALUE)~~ element type ~~name~~indicator for designating an element containing data, and a second ~~(e.g., RECORD)~~ element type ~~name~~indicator for designating an element containing a set of ~~children~~ elements having element type ~~names~~indicators selected from said group.

130. A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding, wherein said message includes plural elements and wherein all elements in said message have element type ~~names~~indicators selected from an encoding group having a predetermined number of ~~element-type-names~~members, including at least a first ~~(e.g., VALUE)~~ element type ~~name~~indicator for designating an element containing data, and a second ~~(e.g., RECORD)~~ element type ~~name~~indicator for designating an element containing a set of ~~children~~ elements having element type ~~names~~indicators selected from said group.

131. A method according to claim any one of claims 127-130, wherein said encoding group further includes a third element type ~~name~~indicator ~~(e.g., LIST or ARRAY)~~ for designating an element containing a set of elements having element type ~~names~~indicators selected from said group.

132. A method according to claim 131, wherein said encoding group includes a fourth element type ~~name~~indicator ~~(e.g., ARRAY or LIST)~~ for designating an element containing a set of elements having element type ~~names~~indicators selected from said group.

133. A method according to claim 131, wherein said encoding group includes a fourth element type ~~name~~indicator ~~(e.g., OBJECT)~~ for designating an element uniquely identifying another encoding element within a particular message.

134. A method according to claim 131, wherein said encoding group includes a fourth element type ~~name~~indicator ~~(e.g., NULL)~~ for designating the absence of a data item.

135. A method according to claim 133, wherein said encoding group includes a fifth element type ~~name~~indicator ~~(e.g., NULL)~~ for designating the absence of a data item.

136. A method according to claim 135, wherein said encoding group includes a sixth element type ~~name~~indicator ~~(e.g., ARRAY or LIST)~~ for designating an element containing a set of elements having element type ~~names~~indicators selected from said group.

137. A method according to claim 131, wherein said third element type ~~name~~indicator ~~(e.g., ARRAY)~~ designates an element containing an n-dimensional array (where n is an integer such that $n \geq 1$) of elements having element type ~~names~~indicators selected from said encoding group.

138. A method according to any one of claims 127-130, wherein said encoding provides a type label associated with an element having said first element type ~~name~~indicator.

139. A method according to claim 138, wherein an element of said first element type ~~name~~indicator with no type label is assumed to be a string type element.

140. A method according to claim 138, wherein said mark-up language is XML, said element type indicators are element type names, and said type label is expressed as an XML attribute on said element having said first element type ~~name~~indicator, with the data type of a data item contained in said element ~~is~~being designated by the value of said attribute.

141. A method according to claim 131, wherein said encoding group further includes a fourth element type ~~name~~indicator ~~(e.g., NUMBER)~~ for designating an element representing a numeric value.

142. A method according to claim 131, wherein said encoding group includes multiple type names each designating a respective different type of data item contained in an element having said first type ~~name~~indicator.

143. A method according to claim 131, wherein said message further includes a semantic label for at least one data item contained in said message.

144. A method according to claim 143, wherein said mark-up language is XML and said semantic label is represented by the value of an XML attribute on the element containing said data item.

145. A method of invoking a service at a first machine from a second machine, said method comprising the steps of:

generating a service invocation request message at said second machine in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group of ~~names~~type indicators, said group ~~of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said message including at least one data item which is a multi-level nested array element where each nesting level corresponds to a respective dimension of said array element; and

transmitting said service invocation request message from said second machine.

146. A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group of ~~names~~type indicators, said group ~~of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality

of data items, where n is an integer and n ≥ 1, said message including at least one data item which is a multi-level nested array element where each nesting level corresponds to a respective dimension of said array element; and

invoking said service in response to said message.

147. A method of invoking a service at a first machine, said method comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group of ~~names~~type indicators, said group ~~of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and n ≥ 1, said message including at least one data item which a multi-level nested array element where each nesting level corresponds to a respective dimension of said array element; and

transmitting said service invocation reply message from said second machine.

148. A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group of ~~names~~type indicators, said group ~~of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and n ≥ 1, said message including at least one data item

which is a multi-level nested array element where each nesting level corresponds to a respective dimension of said array element.

149. A method of invoking a service at a first machine from a second machine, said method comprising the steps of:

generating a service invocation request message at said second machine in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group of ~~names~~type indicators, said group ~~of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said request message including at least one data item which is an array of dimension n and a label associated with said data item and designating said data item as having an array type, said label indicating a value of n but not indicating a size for each of said n dimensions; and

transmitting said service invocation request message from said second machine.

150. A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group of ~~names~~type indicators, said group ~~of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said request message including at least one data item which is an array of dimension n and a label associated with said data item and designating said data item as having an array type, said label indicating a value of n but not indicating a size for each of said n dimensions; and

invoking said service in response to said message.

151. A method of invoking a service at a first machine, said method comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group of ~~names~~type indicators, said group ~~of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said reply message including at least one data item which is an array of dimension n and a label associated with said data item and designating said data item as having an array type, said label indicating a value of n but not indicating a size for each of said n dimensions; and

transmitting said service invocation reply message from said second machine.

152. A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group of ~~names~~type indicators, said group ~~of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said reply message including at least one data item which is an array of dimension n and a label associated with said data item and designating said data item as having an array type, said label indicating a value of n but not indicating a size for each of said n dimensions.

153. A method of invoking a service at a first machine from a second machine, said method comprising the steps of:

generating a service invocation request message at said second machine in compliance with a mark-up language-based message encoding wherein each element in said message is associated with a type indicator selected from a group ~~of names, said group of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said request message including at least one data item which is an array of dimension n and a label associated with said data item and designating said data item as having an array type, said encoding requiring that all data items contained within said array as direct children have the same type as one another; and

transmitting said service invocation request message from said second machine.

154. A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group ~~of names, said group of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said request message including at least one data item which is an array of dimension n and a label associated with said data item and designating said data item as having an array type, said encoding requiring that all data items contained within said array as direct children have the same type as one another; and

invoking said service in response to said message.

155. A method of invoking a service at a first machine, said method comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group ~~of names, said group of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said reply message including at least one data item which is an array of dimension n and a label associated with said data item and designating said data item as having an array type, said encoding requiring that all data items contained within said array as direct children have the same type as one another; and

transmitting said service invocation reply message from said second machine.

156. A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding wherein each element in said message is associated with a type indicator selected from a group ~~of names, said group of names~~ including at least an array type ~~name~~indicator indicating that the corresponding element is an n-dimensional array containing a plurality of data items, where n is an integer and $n \geq 1$, said reply message including at least one data item which is an array of dimension n and a label associated with said data item and designating said data item as having an array type, said encoding requiring that all data items contained within said array as direct children have the same type as one another.

157. A method according to any one of claims 153-156, wherein said label identifies said same type.

158. A method according to any one of claims 149-156, wherein said mark-up language is XML, and said label is expressed as an XML attribute of said element such that the dimension n is given by the value of the attribute

159. A method according to any one of claims 149-156, wherein said message is an XML document.

160. A method according to any one of claims 145-148, wherein said message includes a label associated with said data item and designating said data item as having an array type.

161. A method according to claim 160, wherein each of said second array elements includes at least one data item, with all data items in each of said second array elements being of the same type as one another.

162. A method according to claim 161, wherein said label indicates the type associated with all data items contained in said array.

163. A method according to any one of claims 149-156, wherein said label indicates a value of n but does not indicate a size for each of said n dimensions.

164. A method according to any one of claims 145-148, wherein said message includes a label associated with said data item and designating said data item as having an array type, said encoding requiring that all data items contained within said array as direct children have the same type as one another.

165. A method of invoking a service at a first machine from a second machine, said method comprising the steps of:

generating a service invocation request message at said second machine in compliance with a markup language-based message encoding, wherein each element in

said message is associated with an element type ~~name~~indicator selected from a group ~~of names~~ including at least first and second element type ~~names~~indicators, ~~(e.g., RECORD and OBJECT)~~wherein said message associates an element ~~of~~having said first type ~~name~~indicator with an ID value, and wherein said message includes an element ~~of~~having said second type ~~name~~indicator ~~(OBJECT)~~ which specifies said ID value; and

transmitting said service invocation request message from said second machine.

166. A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding, wherein each element in said message is associated with an element type ~~name~~indicator selected from a group ~~of names~~ including at least first and second element type ~~names~~indicators ~~(e.g., RECORD and OBJECT)~~, wherein said message associates an element ~~of~~having said first type ~~name~~indicator with an ID value, and wherein said message includes an element ~~of~~having said second type ~~name~~indicator ~~(OBJECT)~~ which specifies said ID value; and

invoking said service in response to said message.

167. A method of invoking a service at a first machine, said method comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding, wherein each element in said message is associated with an element type ~~name~~indicator selected from a group ~~of names~~ including at least first and second element type ~~names~~indicators, ~~(e.g., RECORD and OBJECT)~~wherein said message associates an element ~~of~~having said first type ~~name~~indicator with an ID value, and wherein said message includes an element ~~of~~having said second type ~~name~~indicator ~~(OBJECT)~~ which specifies said ID value; and

transmitting said service invocation reply message from said second machine.

168. A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding, wherein each element in said message is associated with an element type ~~name~~indicator selected from a group ~~of names~~ including at least first and second element type ~~names~~indicators, ~~(e.g., RECORD and OBJECT)~~wherein said message associates an element ~~of~~having said first type ~~name~~indicator with an ID value, and wherein said message includes an element ~~of~~having said second type ~~name~~indicator ~~(OBJECT)~~ which specifies said ID value.

169. A method of invoking a service at a first machine from a second machine, said method comprising the steps of:

generating a service invocation request message at said second machine in compliance with a markup language-based message encoding, wherein each element in said message is associated with an element type ~~name~~indicator selected from a group ~~of names~~, said group ~~of names~~ including at least one placeholder element type ~~name~~indicator ~~(e.g., NULL or OBJECT)~~ that designates a placeholder element which represents the absence of data; and

transmitting said service invocation request message from said second machine.

170. A method of invoking a service at a first machine, comprising the steps of:

receiving at said first machine a service invocation request message generated at a second machine in compliance with a markup language-based message encoding, wherein each element in said message is associated with an element type ~~name~~indicator selected from a group ~~of names, said group of names~~ including at least one placeholder

element type ~~name~~indicator ~~(e.g., NULL or OBJECT)~~ that designates a placeholder element which represents the absence of data; and

invoking said service in response to said message.

171. A method of invoking a service at a first machine, said method comprising the steps of:

receiving at said first machine a service invocation request;

invoking said service in response to said request; and

transmitting from said first machine a service invocation reply message in compliance with a markup language-based message encoding, wherein each element in said message is associated with an element type ~~name~~indicator selected from a group ~~of names~~, said group ~~of names~~ including at least one placeholder element type ~~name~~indicator ~~(e.g., NULL or OBJECT)~~ that designates a placeholder element which represents the absence of data; and

transmitting said service invocation reply message from said second machine.

172. A method of invoking a service at a first machine, said method comprising the steps of:

transmitting a service invocation request from a second machine; and

receiving at said second machine a service invocation reply message in compliance with a markup language-based message encoding, wherein , wherein each element in said message is associated with an element type ~~name~~indicator selected from a group ~~of names, said group of names~~ including at least one placeholder element type ~~name~~indicator ~~(e.g., NULL or OBJECT)~~ that designates a placeholder element which represents the absence of data.

173. A method according to any one of claims 169-172, wherein said placeholder element ~~(e.g., NULL)~~ represents a programming language null object reference.